

Random Color Generator Documentation

This document explains how the **Color Generator** works, including the JavaScript code, the logic behind generating random colors, adjusting the text color for visibility, and implementing the copy-to-clipboard functionality.

Table of Contents

- [HTML Structure](#)
- [JavaScript Code Breakdown](#)
 - [Selecting UI Elements](#)
 - [Generating Random Colors](#)
 - [Calculating Brightness](#)
 - [Adjusting Text Color Based on Brightness](#)
 - [Changing the Background Color](#)
 - [Copying the Color Code to Clipboard](#)
 - [Event Listeners](#)
 - [Changing Color on Page Load](#)
- [Conclusion](#)

HTML Structure

The HTML contains the following key elements:

- A `div` with the ID `#color-box` to display the color.
- A button with the ID `#change-color-btn` to change the color.
- A `span` with the ID `#color-code` to display the generated color code.
- A button with the ID `#copy-btn` to copy the color code.
- A hidden `div` with the ID `#copy-feedback` to show feedback when the color code is copied.

Sample HTML

```
<div id="color-box" class="color-box"></div>
<span id="color-code"></span>
<button id="change-color-btn">Change Color</button>
<button id="copy-btn">Copy Color Code</button>
<div id="copy-feedback" class="hidden">Copied!</div>
```

JavaScript Code Breakdown

The javascript contains the following:

Selecting UI Elements

I used `document.querySelector()` to select various HTML elements so we can interact with them in the JavaScript code.

```
const colorBox = document.querySelector("#color-box");
const changeColorBtn = document.querySelector("#change-color-btn");
const colorCode = document.querySelector("#color-code");
const copyBtn = document.querySelector("#copy-btn");
const copyFeedback = document.querySelector("#copy-feedback");
```

Generating Random Colors

The function `getRandomColors()` generates a random hex color. Hex colors always begin with `#` followed by 6 characters (numbers from 0–9 or letters from A–F).

I looped 6 times to pick random characters from 0123456789ABCDEF to generate the color code.

```
function getRandomColors() {
  const colorLetters = "0123456789ABCDEF"; // Hexadecimal characters
  let color = "#"; // Colors start with '#'

  // Loop 6 times to generate the 6 characters for the color code
  for (let i = 0; i < 6; i++) {
    color += colorLetters[Math.floor(Math.random() * 16)];
  }

  return color; // Return the random color code
}
```

Calculating Brightness

The `getBrightness()` function calculates the brightness of the generated color. This helps us decide if the text inside the box should be black or white for better readability.

I extracted the Red, Green, and Blue (RGB) values from the hex code. The brightness is calculated using the formula:

$$(R * 299 + G * 587 + B * 114) / 1000$$

```
function getBrightness(hexColor) {
  hexColor = hexColor.replace("#", ""); // Remove '#' symbol

  // Convert the hex parts to RGB values
  const r = parseInt(hexColor.substring(0, 2), 16);
  const g = parseInt(hexColor.substring(2, 4), 16);
  const b = parseInt(hexColor.substring(4, 6), 16);

  // Return the brightness using a common formula
```

```
    return (r * 299 + g * 587 + b * 114) / 1000;  
}
```

Adjusting Text Color Based on Brightness

The `adjustTextColor()` function sets the text color based on the brightness of the background color:

If the background is bright (brightness > 150), the text color is set to black. Otherwise, the text color is set to white.

```
function adjustTextColor(hexColor) {  
    const brightness = getBrightness(hexColor);  
    colorCode.style.color = brightness > 150 ? "#000000" : "#FFFFFF";  
}
```

Changing the Background Color

The `changeColor()` function changes the background color of the box and updates the displayed color code:

- It calls `getRandomColors()` to get a new color.
- It updates the background color of the `#color-box`.
- It displays the new color code inside `#color-code`.
- It adjusts the text color using `adjustTextColor()`.

```
function changeColor() {  
    const newColor = getRandomColors(); // Generate new random color  
    colorBox.style.backgroundColor = newColor; // Change background color  
    colorCode.innerText = newColor; // Display the color code in the box  
    adjustTextColor(newColor); // Adjust the text color for visibility  
}
```

Copying the Color Code to Clipboard

The `copyToClipboard()` function allows users to copy the color code to their clipboard:

- A temporary textarea element is created and added to the document.
- The color code is selected and copied using `document.execCommand("copy")`.
- The textarea is then removed from the document.
- A "Copied!" message is displayed for 1.5 seconds after copying.

```
function copyToClipboard(text) {  
    const textarea = document.createElement("textarea"); // Create a temporary  
    text area  
    textarea.value = text; // Set its value to the color code  
    document.body.appendChild(textarea); // Add it to the page  
    textarea.select(); // Select the text
```

```
document.execCommand("copy"); // Copy it to the clipboard
document.body.removeChild(textarea); // Remove the text area

// Show "Copied!" message for 1.5 seconds
copyFeedback.classList.remove("hidden");
setTimeout(() => {
  copyFeedback.classList.add("hidden");
}, 1500);
}
```

Event Listeners

We set up two event listeners:

- One for the "Change Color" button to change the background color when clicked.
- Another for the "Copy Color Code" button to copy the color code to the clipboard when clicked.

```
changeColorBtn.addEventListener("click", changeColor); // When "Change Color"
button is clicked
copyBtn.addEventListener("click", () => {
  copyToClipboard(colorCode.innerText); // When "Copy Color Code" button is
clicked
});
```

Changing Color on Page Load

When the page first loads, we call `changeColor()` to set an initial random color.

```
changeColor(); // Set a random color on page load
```

Conclusion

This JavaScript code enables a dynamic color generator that:

- Generates random colors for the background.
- Adjusts the text color for better visibility.
- Allows users to copy the generated color code to the clipboard.
- The functions are structured to handle both the generation of random colors and user interactions efficiently.